Алгоритм Yarrow — краткий обзор

Ситкарев Г.А. <sitkarev@komitex.ru>

Сыктывкарский Государственный Университет
Лаборатория Программирования и Прикладной Математики
http://amplab.syktsu.ru

1. Компоненты Yarrow

Yarrow это криптографически стойкий генератор псевдослучайных чисел. Он состоит из четырёх основных компонентов:

1. Аккумулятора энтропии, который собирает семплы из источников энтропии в два пула (быстрый и медленный).
2. Механизма пересева, периодически пересевающего ключ новой энтропией из пулов.
3. Механизма генерации, выполняющего генерацию псевдослучайной последовательности из ключа.
4. Управления пересевом, определяющего когда нужно пересевать ключ.

2. Аккумулятор энтропии

Аккумулятор энтропии состоит из двух пулов:

1. Быстрый пул (fast pool) обеспечивает частые пересевы ключа.
2. Медленный пул (slow pool) обеспечивает редкие, но очень консервативные (по оценке имеющейся в нём энтропии) пересевы ключа.

Даже если наши предположения об энтропии оказались слишком оптимистичными, мы всё равно рано или поздно выполним защищённый пересев ключа из медленного пула. Каждый из пулов представляет собой текущее значение односторонней хеш-функции, на вход которой подаются семплы, содержащие энтропию. Хеш-функция выполняет роль сборщика энтропии. Семплы из источников поочерёдно поступают то в быстрый то в медленный пул.

2.1. Измерение энтропии

Для каждого источника пул поддерживает оценку полученной энтропии в битах. Измерение энтропии весьма непростой процесс, однако практически предлагается использовать три способа:

1. Значение энтропии каждого семпла передаётся программистом в виде коэффициента.
2. Для каждого источника постоянно выдаётся оценка энтропии через специализированный определитель. Практически целесообразно для обнаружения случаев, когда источник начинает выдавать совсем низкую энтропию.
3. Общесистемный лимит, константа (0,5 в Yarrow).

Для оценки энтропии каждого семпла используется наименьший из них.

3. Механизм пересева

Механизм пересева соединяет аккумулятор энтропии и механизм генерации. Когда механизм пересева определяет, что нужно осуществить пересев ключа, компонента пересева должна обновить ключ, используемый механизмом генерации, из одного или двух пулов. Осуществить это нужно так, что если атакующему был известен или ключ или содержимое пулов, то ключ пересева остался ему неизвестным.

Для генерации нового ключа пересев из быстрого пула использует текущий ключ и хеш всех входов в быстрый пул с момента последнего пересева. После этого оценка энтропии для всех источников в быстром пуле сбрасывается в 0.

Для генерации нового ключа пересев из медленного пула использует текущий ключ, хеш всех входов в быстрый пул и хеш всех входов в медленный пул. После этого оценка энтропии для всех источников в быстром и медленном пулах сбрасывается в 0.

4. Управление пересевом

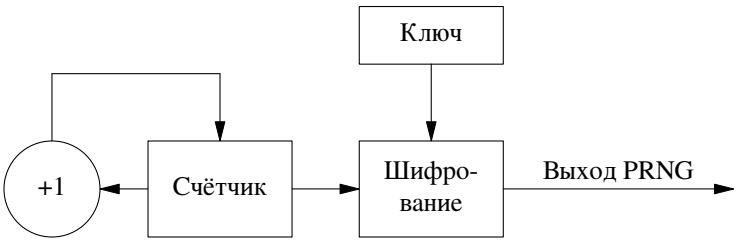
Частый пересев желателен, но потенциально повышает вероятность успеха итерационной атаки. Редкий пересев оставляет больше информации атакующему, получившему ключ. Дизайн управления пересевом должен быть компромиссным между этими двумя взаимно противоречивыми требованиями.

Для каждого источника энтропии поддерживается счётчик. Когда любой источник превысил порог (100 бит), выполняется пересев из быстрого пула. Когда k источников превысят порог (160 бит), выполняется пересев из медленного пула. Практически целесообразно, имея хорошие источники энтропии, выбрать $k = 3$.

5. Механизм генерации

Нам нужны два алгоритма:

- Односторонняя хеш-функция $h(x)$, с отпечатком размером m–бит.
- Блочный шифр $E_K()$ с ключом длиной k–бит и n–битным блоком.



Для генерации следующих n–бит псевдослучайной последовательности счётчик C увеличивается на 1 и зашифровывается блочным шифром, используя ключ K:

$$C \leftarrow C + 1 \bmod 2^n$$
$$R \leftarrow E_K(C)$$

где R — выход псевдослучайной последовательности, а K — текущий ключ.

Если в какой-то момент ключ оказался скомпрометирован, нужно предотвратить утечку предыдущих выходных значений, которые атакующий может получить. Описанный механизм не имеет такой защиты от утечки, поэтому дополнительно ведётся подсчёт количества блоков псевдослучайных последовательностей, выданных на выход. Как только будет достигнут некий лимит $1 \le P_g \le 2^{n/3}$, генерируется k–битовый выход и устанавливается как ключ.

$$K \leftarrow \text{следующие } k \text{ бит выхода PRNG}$$

Следует выбирать весьма консервативное значение P_g, в Yarrow оно задано как 10.

6. Механизм пересева

Механизм пересева генерирует новый ключ K для генератора из энтропии, собранной в аккумуляторе энтропии, и существующего ключа. Время выполнения механизма пересева зависит от параметра $P_t \ge 0$.

Механизм пересева состоит из следующих шагов:

1. Аккумулятор энтропии вычисляет хеш всех входов в быстрый пул. Результат этого вычисления назовём v_0.
2. Установить $v_i := h(v_{i-1} | v_0 | i)$, для $i = 1,\ldots,t$.
3. Установить $K \leftarrow h'(h(v_{P_t} | K), k)$.
4. Установить $C \leftarrow E_K(0)$.
5. Сбросить все счётчики энтропии в пулах в 0.
6. Очистить всю память, задействованную под хранение промежуточных значений.

7. Если используется файл посева ключа, то следующие $2k$ бит выхода псевдослучайной последовательности перезаписывают содержимое этого файла.

Функция $h'(m, k)$ определяется в терминах $h(x)$ следующим образом:

$$\begin{aligned} s_0 &:= m \\ s_i &:= h(s_0|\ldots|s_{i-1}), \quad i = 1,\ldots \\ h'(m, k) &:= \text{первые } k\text{–бит } (s_0|s_1|\ldots) \end{aligned}$$

Она является «адаптором размера» входного сообщения любой длины в выходное сообщение заданной длины. Если длины входного и выходного сообщений совпадают, то функция выдаёт входное сообщение (без изменений) на выход.